Escuela Técnica Superior de Ingeniería (ICAI)
Departamento de Electrónica y Automática

Lab-3: INTRODUCCIÓN A LA SIMULACIÓN Y A LA IMPLANTACIÓN FÍSICA CON QUARTUS II

OBJETIVOS

En esta práctica, además de utilizar la herramienta para diseñar hardware digital *Quartus II*, del fabricante *Altera*, se empezará a usar una herramienta de simulación de circuitos y sistemas digitales, llamada *ModelSim* del fabricante *Mentor Graphics*.

Al finalizar la práctica, el alumno debe ser capaz de:

- Simular un circuito digital usando *Quartus II y ModelSim*.
- Volcar y configurar el diseño en un dispositivo de lógica programable FPGA. Verificando su funcionamiento.

MATERIAL

- Ordenador personal con *Quartus II y ModelSim*.
- Tarjeta de desarrollo de lógica programable.

DURACIÓN

1 sesión.

TRABAJO PREVIO E INFORMACIÓN DE CONSULTA

En esta sesión se harán dos tareas previas:

- Leer el *Manual de Usuario de la Placa de Lógica Programable DE1* para familiarizarse con la tarjeta de desarrollo de lógica programable del laboratorio.
- Leer el enunciado de la práctica.
- Llevar al laboratorio todos los ficheros del proyecto de la práctica anterior.

TRABAJO POSTERIOR Y ENTREGA DE RESULTADOS

Después del desarrollo de la práctica se debe entregar un informe de resultados (véase guía de informe) y el trabajo desarrollado en el laboratorio en soporte informático que contenga el proyecto completo y que incluye la simulación de esta práctica.

INTRODUCCIÓN

Aunque no se le vea utilidad a la fase de simulación del circuito capturado en la práctica anterior, es muy importante simular los sistemas que se desarrollen.

En este caso tan simple se podría descargar el diseño en la tarjeta del laboratorio para comprobar su funcionamiento. Sin embargo, la simulación presenta varias ventajas:

- Permite seguir fácilmente la evolución temporal de todas las señales. Para conseguir ver el mismo número de señales en el circuito real necesitaríamos un analizador lógico, que es más caro, voluminoso y delicado.
- Además de las entradas y salidas del circuito se pueden analizar señales internas, lo que permite depurar el funcionamiento del circuito más a fondo.
- También se puede simular sólo una parte del circuito. Por ejemplo, se podría haber simulado el componente comparador sin necesidad de haber hecho el esquema superior (Pract2). Esto permite ir probando cada bloque jerárquico del circuito conforme se va construyendo, con lo que la depuración es mucho más fácil al tener que centrarse cada vez sólo en una pequeña parte del circuito, en lugar de enfrentarse al circuito completo.
- Permite corregir errores en el diseño y comprobar el funcionamiento del mismo sin necesidad de disponer del hardware.

Sin embargo no son todo ventajas. Los principales inconvenientes de la simulación frente a la implantación física son:

- No se dispone de los elementos de salida. En esta práctica no es muy importante. Pero sí por ejemplo, si quisiéramos comprobar un circuito de procesamiento de audio, sería más fácil verificar su salida, escuchando el sonido que viendo muestras en una pantalla.
- La especificación de la evolución temporal de las señales de entrada puede ser tediosa,
- En circuitos complejos, la simulación puede tardar demasiado tiempo en ejecutarse (aunque este caso no se da en este curso).

La cuarta fase del desarrollo consiste en volcar el diseño realizado y simulado, en la FPGA. Este volcado no es más que la escritura del conexionado interno de las macroceldas de la FPGA entre sí, y la configuración de las mismas vistas en la práctica anterior.

Antes de comenzar a trabajar, no olvide crear el directorio de trabajo para la práctica de esta sesión y copiar los archivos generados en la práctica del otro día.

DESARROLLO PRÁCTICO

En primer lugar, hay que copiar todos los archivos generados en la práctica anterior al directorio de trabajo de esta práctica.

A continuación, hay que arrancar Quartus II 10.1 que está en la ruta:

`Botón de Inicio → Programas → Altera → Quartus II 10.1 Web Edition.`

A continuación, hay que abrir el proyecto declarado en la práctica anterior (`Pract2.qpf`).Para ello, en el menú `File`, hay que escoger la opción `Open Project…` y seleccionar el archivo `Pract2.qpf` de la carpeta de trabajo de hoy.

SIMULAR EL PROYECTO COMPILADO

Antes de implantar físicamente el circuito en la FPGA, se va a simular su funcionamiento con ayuda del entorno *Quartus II* y *ModelSim*.

GENERACIÓN DE FICHERO DE VECTORES DE PRUEBA

Para simular cualquier sistema o circuito, es necesario generar el conjunto de patrones de las señales de entrada que permiten evaluar el funcionamiento de nuestro sistema digital lo más completo posible. La generación de este conjunto de patrones es conocido como vectores de prueba o banco de pruebas (*testbench*). El simulador *ModelSim* está preparado para proporcionar estos patrones de prueba a través de un fichero VHDL. Este fichero de pruebas se genera en el *QuartusII* después de haber compilado el diseño de nuestro circuito digital. Este fichero hay que completarlo añadiendo una descripción de cómo queremos que evolucionen en el tiempo las señales de entrada o de excitación. A partir de este fichero, mi circuito es capaz de simular su comportamiento y verificar que la evolución de las señales es correcta. Para ello, se introducen los *vectores de prueba*, es decir las señales de entrada y su temporización, que permitan comprobar el bloque sujeto a simulación

Para generar este fichero, vaya al menú (véase Figura 1).

`Processing → Start → Start Test Bench Template Writer`

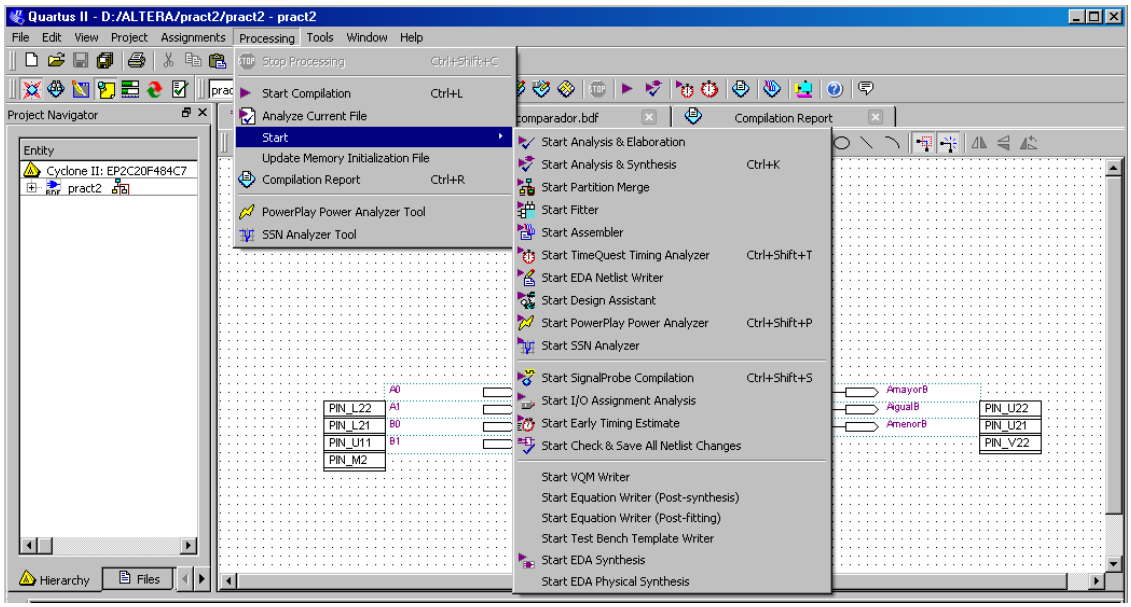

Figura 1. Generación del fichero de vectores de prueba (testbench).

La plantilla de este fichero se crea en la carpeta de trabajo bajo el directorio *simulation/modelsim/pract2.vht*

VECTORES DE PRUEBA

El proceso siguiente a la generación de este fichero de vectores de prueba es la edición de los estímulos que nos permitan simular el circuito. Antes de esto vamos a ver las características de este fichero. Tiene el aspecto del texto que a continuación se presenta en donde se pueden ver varios bloques:

- Un bloque de librerías empleadas en la simulación.
- Un bloque de entidad con el nombre dado por defecto: `ENTITY pract2_vhd_tst.`
- Un bloque de arquitectura con las señales de entrada y salida de nuestro circuito a simular.
- Los componentes a simular (`pract2`), al que se le asigna la instancia (`i1`) y sus conexiones.
- Finalmente los procesos (`PROCESS`) que contienen los vectores de prueba. Es esta sección la que anotaremos cómo nuestras señales de entrada de mueven para simular todos los casos posibles que permitan comprobar nuestro circuito.

Es importante obtener una combinación lo más completa posible de las señales de entrada para verificar que el circuito funciona correctamente bajo cualquier circunstancia.

```
-- ********************************************************************************
-- This file contains a Vhdl test bench template that is freely editable to
-- suit user's needs .Comments are provided in each section to help the user
-- fill out necessary details.
-- ********************************************************************************
-- Generated on "01/19/2011 18:22:28"

-- Vhdl Test Bench template for design  :  pract2
--
-- Simulation tool : ModelSim-Altera (VHDL)

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY pract2_vhd_tst IS
END pract2_vhd_tst;

ARCHITECTURE pract2_arch OF pract2_vhd_tst IS
-- constants
-- signals
SIGNAL A0 : STD_LOGIC;
SIGNAL A1 : STD_LOGIC;
SIGNAL AigualB : STD_LOGIC;
SIGNAL AmayorB : STD_LOGIC;
SIGNAL AmenorB : STD_LOGIC;
SIGNAL B0 : STD_LOGIC;
SIGNAL B1 : STD_LOGIC;

COMPONENT pract2
	PORT (
	A0 : IN STD_LOGIC;
	A1 : IN STD_LOGIC;
	AigualB : OUT STD_LOGIC;
	AmayorB : OUT STD_LOGIC;
	AmenorB : OUT STD_LOGIC;
	B0 : IN STD_LOGIC;
	B1 : IN STD_LOGIC
	);
END COMPONENT;

BEGIN
	i1 : pract2
	PORT MAP (
-- list connections between master ports and signals
	A0 => A0,
	A1 => A1,
	AigualB => AigualB,
	AmayorB => AmayorB,
	AmenorB => AmenorB,
	B0 => B0,
	B1 => B1
	);

init : PROCESS
-- variable declarations
BEGIN
	 -- code that executes only once
WAIT;
END PROCESS init;

always : PROCESS
-- optional sensitivity list
-- (        )
-- variable declarations
BEGIN
	-- code executes for every event on sensitivity list
WAIT;

END PROCESS always;
END pract2_arch;
```

Con este fichero estamos ya listos para generar el proceso de edición de vectores de prueba y simulación.

EDITANDO EL FICHERO DE TEST

Para que la simulación tenga efecto, es necesario introducir los estímulos en el fichero de test (prac2.vht). Para ello vamos a editarlo desde el *QuartusII*.

`File → Open → clic en directorio simulation → clic en modelsim`

Elegir tipo de fichero: `Test Bench Output Files (*.vht *.vt)`

Seleccionar `pract2.vht`

El fichero se abre en la ventana de edición de *QuartusII*. Se puede editar entonces los estímulos y su forma de onda para simular el circuito digital. El fichero generado lo podéis ver completo en el apartado anterior. Para esta primera simulación, editaremos el fichero e introduciremos unos cuantos vectores de prueba para los valores de A y B, con el objeto de hacer todas las combinaciones posibles. Es decir, para comprobar el circuito, le diremos **en forma textual** que la señales A0, A1, B0 y B1, cambien en el tiempo de la forma en que se indica en el diagrama de la Figura 2.

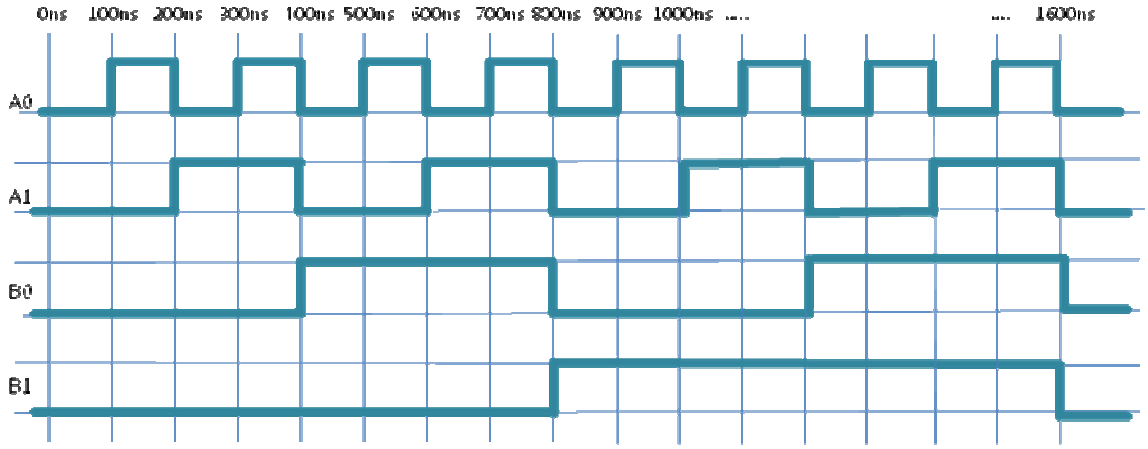

Figura 2. Gráfico de los vectores de prueba para *test bench*

Para ello, y como estamos editando el fichero pract2.vht, introducimos estos valores en la sección *PROCESS* siguiente. Cada 100 nanosegundos haremos que cambie de valor una de las señales de forma que se representen todas las combinaciones posibles y en consecuencia ver el valor de las salidas para todas las posibles combinaciones a la entrada. Termine de introducir los vectores de prueba según el diagrama y guarde el fichero con el mismo nombre.

```
always : PROCESS
-- optional sensitivity list
-- (        )
-- variable declarations
BEGIN
        -- code executes for every event on sensitivity list
       A0 <= '0'; A1 <= '0'; B0 <= '0'; B1 <= '0';
       wait for 100 ns;
       A0 <= '1';
       wait for 100 ns;
       A0 <= '0'; A1 <= '1';
       wait for 100 ns;
       A0 <= '1';
       wait for 100 ns;
       A0 <= '0'; A1 <= '0'; B0 <= '1';
       wait for 100 ns;
       A0 <= '1';
       wait for 100 ns;
       A0 <= '0'; A1 <= '1';
       wait for 100 ns;
       A0 <= '1';
       wait for 100 ns;
       A0 <= '0'; A1 <= '0'; B0 <= '0';
       wait for 100 ns;
       A0 <= '1';
       wait for 100 ns;
       A0 <= '0'; A1 <= '1';
       wait for 100 ns;
       A0 <= '1';
       wait for 100 ns;
       A0 <= '0'; A1 <= '0'; B0 <= '0'; B1 <= '1';
       wait for 100 ns;
       A0 <= '1';
       wait for 100 ns;

WAIT;
```

El paso para simular es ir al menú principal:

`Assignments → Settings → (Pulsar) Simulation`

Lo que nos despliega el cuadro de dialogo siguiente, Figura 3.

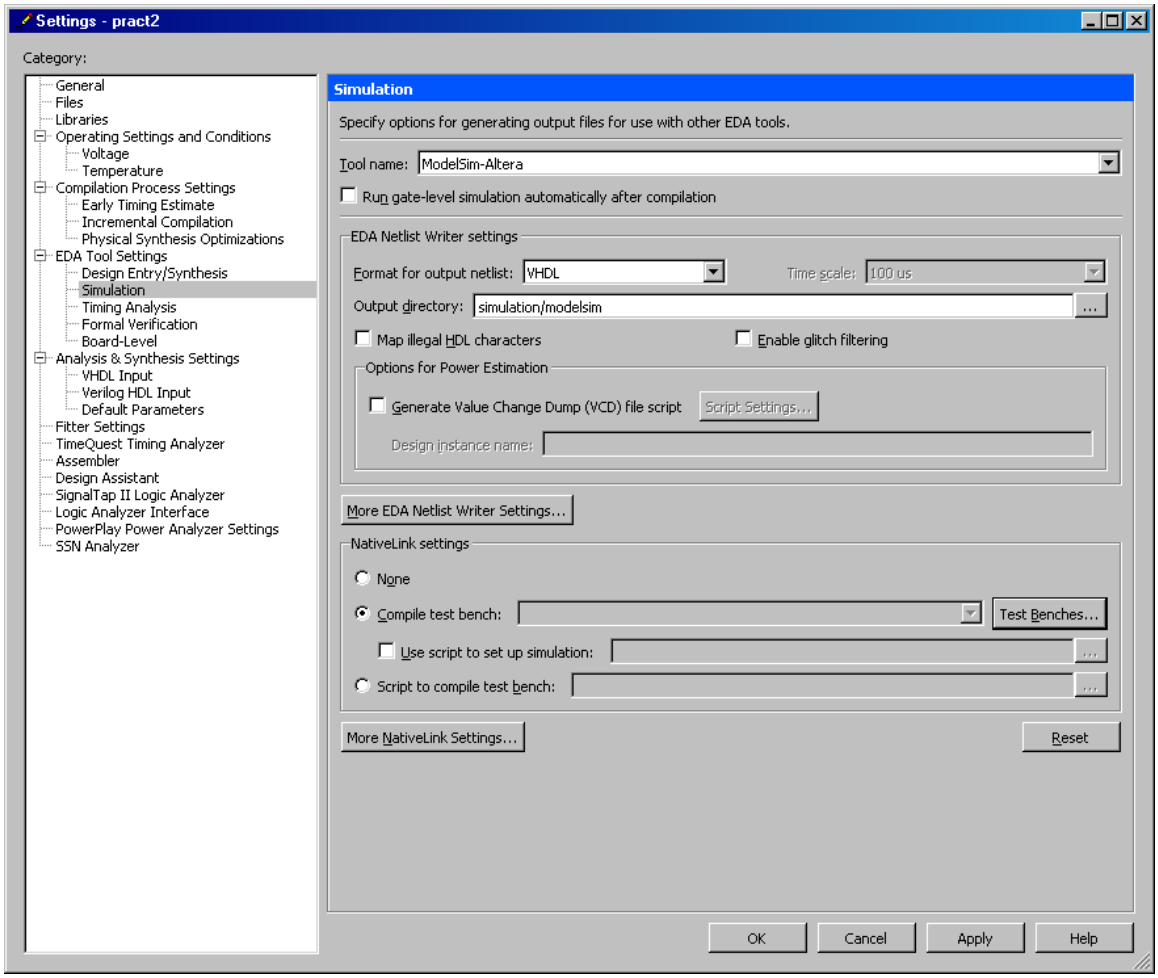

Figura 3. Cuadro de diálogo para compilación del *testbench*.

Debe estar indicado en el recuadro `Tool name`, la opción `ModeSim-Altera`. Si ha hecho la generación del proyecto como se ha indicado, estará correcto, sino abra el recuadro y ponga esta opción. A continuación haga clic en la opción `Compile test bench` y haga clic en `Test Benches`…. Se nos abre otro cuadro de dialogo, que nos permite seleccionar el fichero de test que vamos a usar. Clic en `New`. Este último paso nos abre la siguiente ventana Figura 4. Ha de rellenar los campos mostrados en la figura, teniendo especial cuidado en poner los nombres del *test bench* y de la instancia (i1) correctamente.

`Test bench name`: es el nombre de la entidad que aparece al principio del fichero.

Ha de seleccionar "`Use test bench to perfom VHDL timing simulation`" y en el recuadro " "`Design instance name in test bench`" poner **i1**.

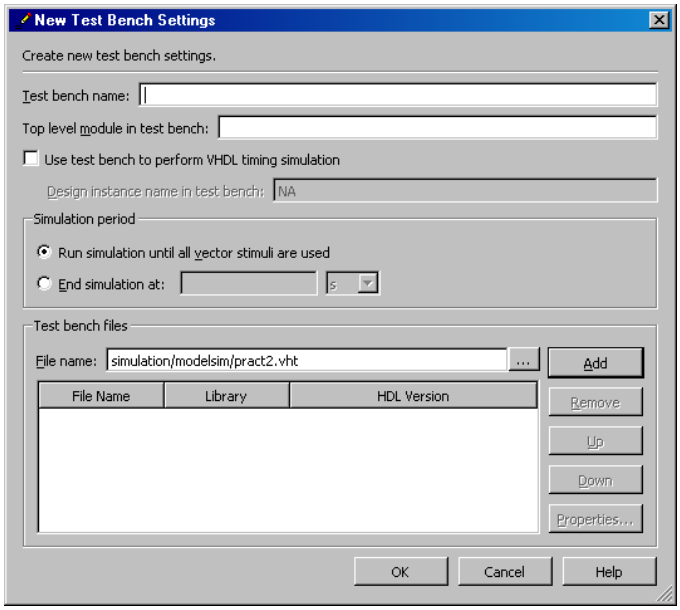

Figura 4. Cuadro de diálogo seleccionar el fichero de *testbench*

Ponga el camino en donde se encuentra el fichero de test bench en la sección `Test bench files`. Para ello abra el botón … y bajo el directorio `simulation/modelsim`, seleccione el fichero. (`pract2.vht`). A continuación clic en `Add`. Esto hace que aparezca el fichero en la lista del recuadro inferior. Figura 5.

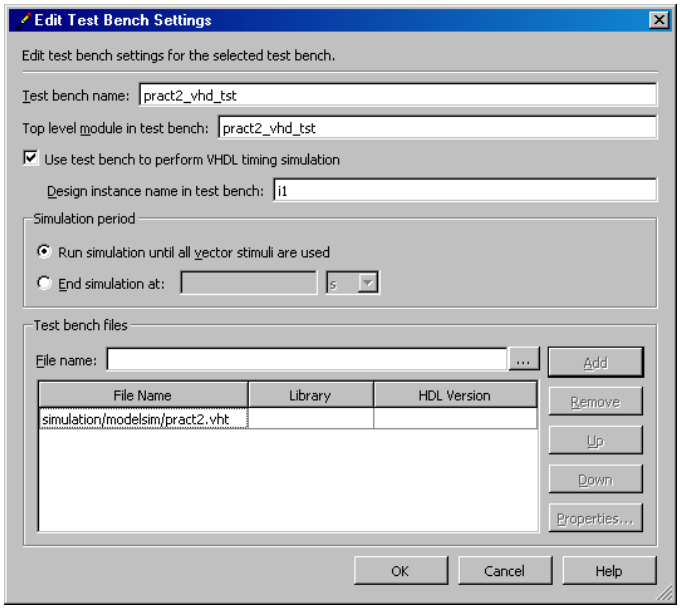

Figura 5. Fichero y cuadro de configuración de *testbench*

Rellene el cuadro de la Figura anterior tal como se ha indicado. El parámetro `Test bench name`, en la primera línea poniendo el nombre del fichero `pract2_vhd_tst`. Clic el recuadro `Use test bench to perfom VHDL timing simulation` y ponga el número de la instancia (i1) en la línea. El cuadro queda

rellene tal como la Figura. Para finalizar clic OK. Para volver a la ventana anterior con los parámetros del test y del fichero de simulación se abre, Figura 6. Dar OK.

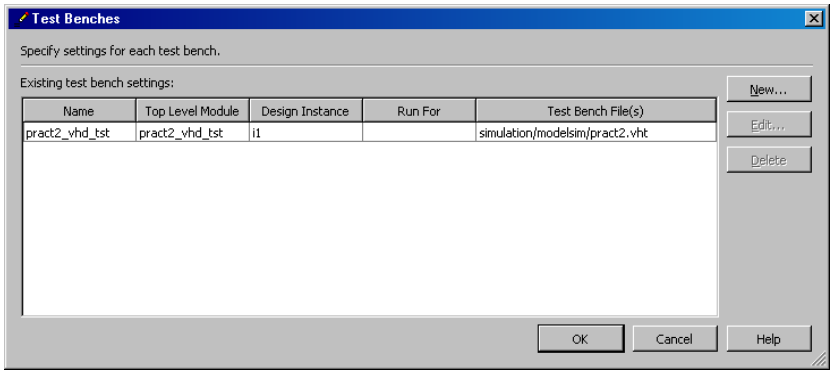

Figura 6. Fichero y configuración de *testbench.*

Se termina la configuración de la simulación dando OK en el recuadro de `Simulation`. Figura 7.

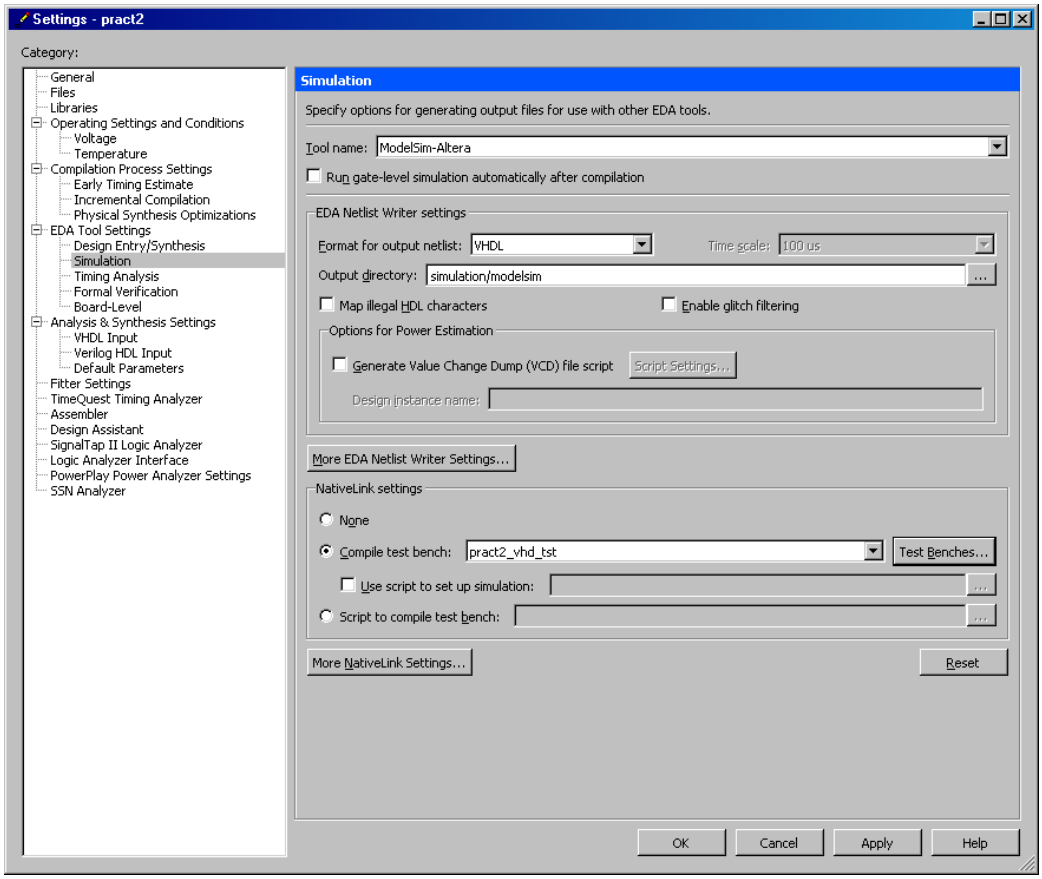

Figura 7. Configuración de simulador con el fichero de *testbench*

Habiendo terminado de editar y configurar, procedemos a invocar el simulador *ModelSim*. Introduzca en el menú principal la opción `Tools` → `Run EDA Simulation Tool` → `EDA Gate Level Simulation` (Figura 8). Aparece entonces la ventana de la derecha en la figura. Clic en `Run`.

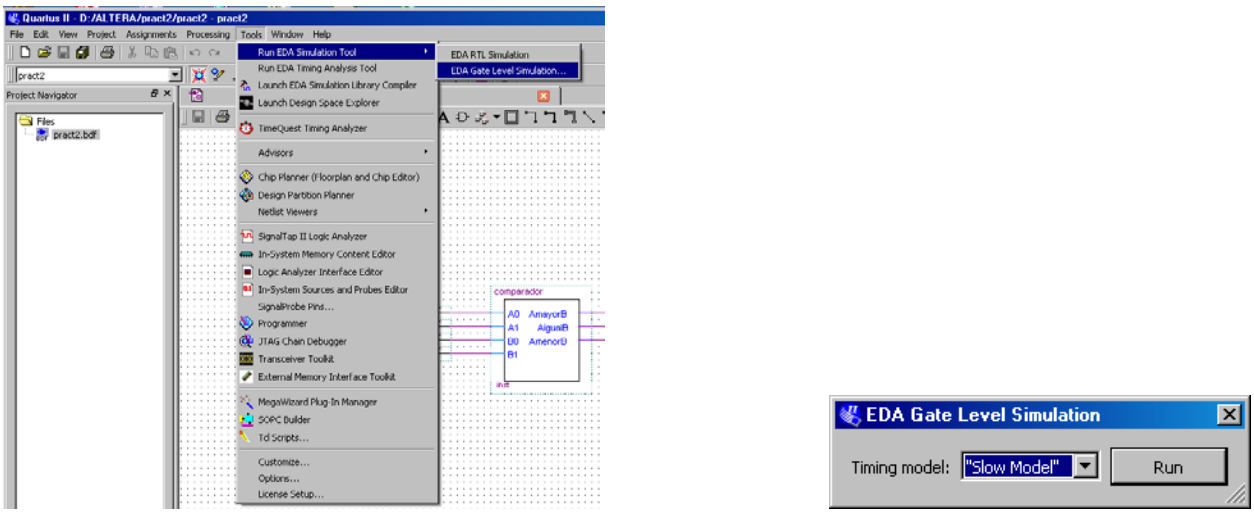

Figura 8. Invocando al simulador.

Este proceso arranca el simulador *ModelSim* con el fichero de *testbench* y una serie de ventanas en el simulador, según este configurado el menú `View` (en el menú principal). La vista clásica es como se refleja en la Figura 9. En esta figura se ven tres ventanas, la primera de ellas indica la librería que componentes que se simulan (arriba izquierda). Otra más con objetos, que representan señales del circuito (abajo izquierda) y la de formas de onda (*Wave*) a la derecha de la imagen.

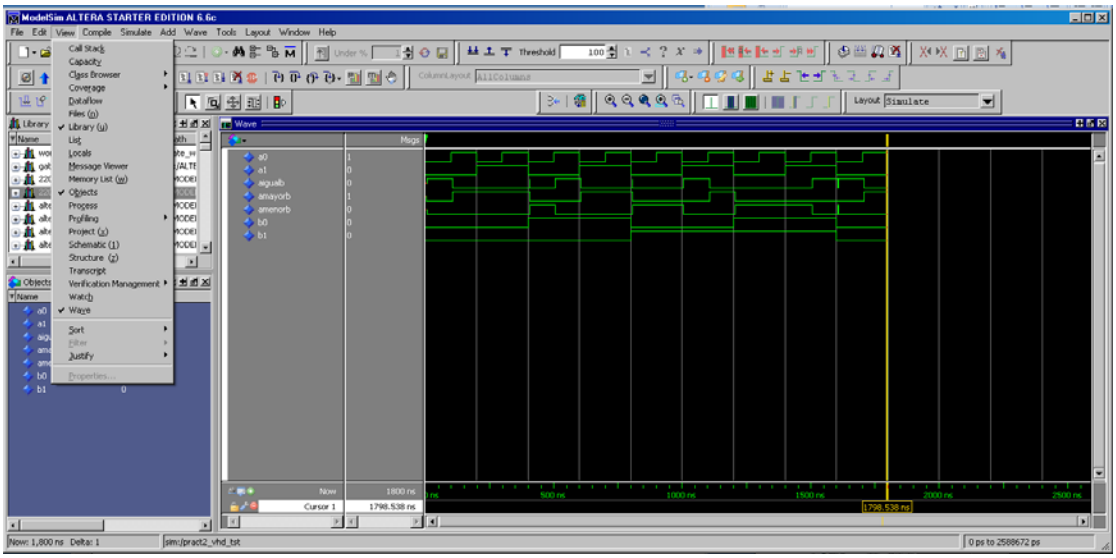

Figura 9. Vista del *Modelsim*, ventanas activas-

La ventana más importante que nos interesa comprobar es la ventana de forma de onda (*Wave*). En ella se reflejan las entradas y salidas del circuito simulado. Las entradas tienen el aspecto que se les ha dado en el fichero de *test bench*. En la Figura 9, también puede verse un conjunto de barras con iconos

(*Toolbar*). Estas barras de iconos se quitan o ponen, seleccionando en el menú principal *Window* → *Toolbars*.

SIMULANDO Y VISUALIZANDO SEÑALES

Para lanzar una nueva simulación, se selecciona desde el menú principal en `Simulate → Start simulation`. Esta opción abre una ventana para seleccionar el circuito y testbench que queremos simular. `Seleccione gate_work → pract2_vhd_tst.` Clic OK. Lo que reinicia el simulador. Figura 10.

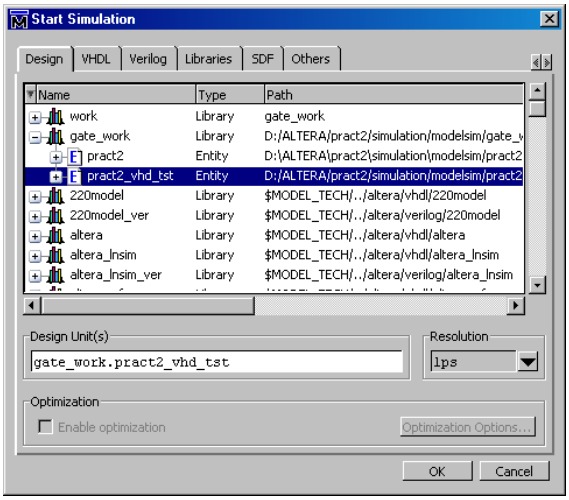

Figura 10. Arranque del simulador desde *ModelSim*.

En este momento, se arranca el simulador con las ventanas y barras de herramientas seleccionadas. En la ventana *Wave* vamos a añadir las señales que queremos simular. Para ello, seleccione de la ventana de objetos, las señales de entrada y salida y arrástrelas a la ventana *Wave*. O, haga clic con el botón derecho del ratón y seleccione la opción `Add→To wave→Selected signals`. Aparecen ahora las señales en la ventana `Wave`. Estamos ya listos para lanzar los vectores de simulación, vaya a: `Simulate→Run-All.` Esta opción lanza todos los vectores de test del fichero de *testbench* durante el tiempo que se ha especificado en el fichero. Figura 11.

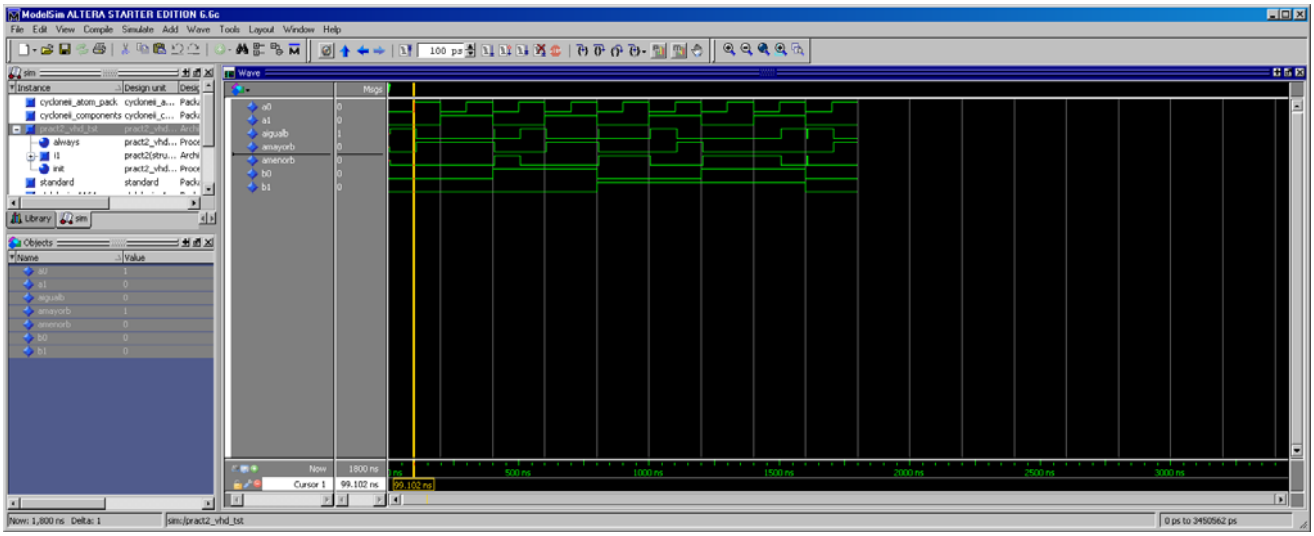

Figura 11. Simulando desde *ModelSim*.

La barra de herramientas de simulación contiene estas mismas opciones, en los iconos siguientes:

. En éste, se pueden observar a la izquierda del cuadro de 100 ps, la opción `Restart`, que permite reinicar la simulación. A continuación `Run`, que lanza la simulación el tiempo indicado en el recuadro, a continuación `ContinueRun, Run-All y Break`.

Una vez completada la simulación se ha de verificar si el circuito se comporta correctamente, comprobando que las salidas cumplen la tabla de verdad del anexo de la práctica.

EDITANDO EL FICHERO DE ESTIMULOS Y COMPILANDO

El proceso normal que sigue el simular un circuito, es un proceso iterativo de comprobación y depuración, por lo que es normal que se necesite cambiar, ampliar o modificar los estímulos a la simulación, bien sea por cambios en el circuito a simular o por visualizar otras señales, etc. Para ello, podemos editar el fichero de estímulos desde el *ModelSim*, yendo a `File → Open`, abre un cuadro de diálogo, donde tenemos que seleccionar el tipo de fichero `Tipo→VHDL Files`. Aparece entonces *pract2.vht*. Seleccionamos este fichero. Aparece una nueva ventana con el fichero de *textbench* para ser editado. Se ejecutan los cambios o modificaciones, en este ejemplo, cambie el tiempo entre las transiciones de las señales, en lugar de 100ns. ponga 200 ns., guarde el fichero.

Antes de volver a simular, es necesario recompilar el fichero antes editado. Se puede hacer, desde el editor del fichero, en la barra de herramientas con el icono de compilación, o desde la ventana Library haciendo clic en el botón derecho del ratón lo que abre una ventana con la opción `Recompile`. Ver Figura 12.

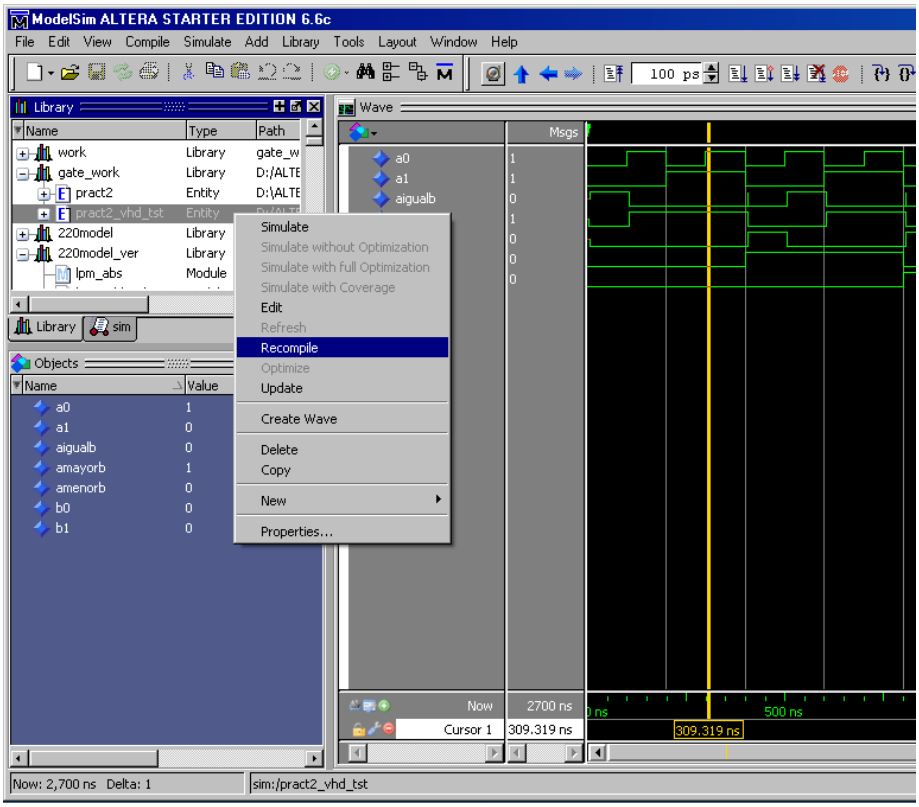

Figura 12. Recompliando un fichero de testbench

Recompilado el fichero de test, se reinicia el simulador. Esto se puede hacer con el botón o haciendo clic en `Simulate→Run→Restart`. Estamos ya listos para volver a simular, haciendo clic en el botón `Run` o haciendo clic en `Simulate→Run→Run-All`. Proceda cuando termine a guardar las simulaciones de sus diseños.

CONFIGURAR LA FPGA CON EL CIRCUITO COMPILADO

Una vez que ha finalizado la simulación y que se ha comprobado que el circuito se comporta perfectamente, se puede pasar a su implantación física en la FPGA.

En primer lugar es necesario conectar la tarjeta de desarrollo de lógica programable del laboratorio al ordenador mediante el cable **USB.** Después hay que asegurarse de que el interruptor **RUN/PROG** que hay a la izquierda de la tarjeta está en **RUN** (véase la Figura 17).

Por último, hay que encender la tarjeta pulsando el interruptor rojo que hay en la parte superior izquierda (véase la Figura 13).

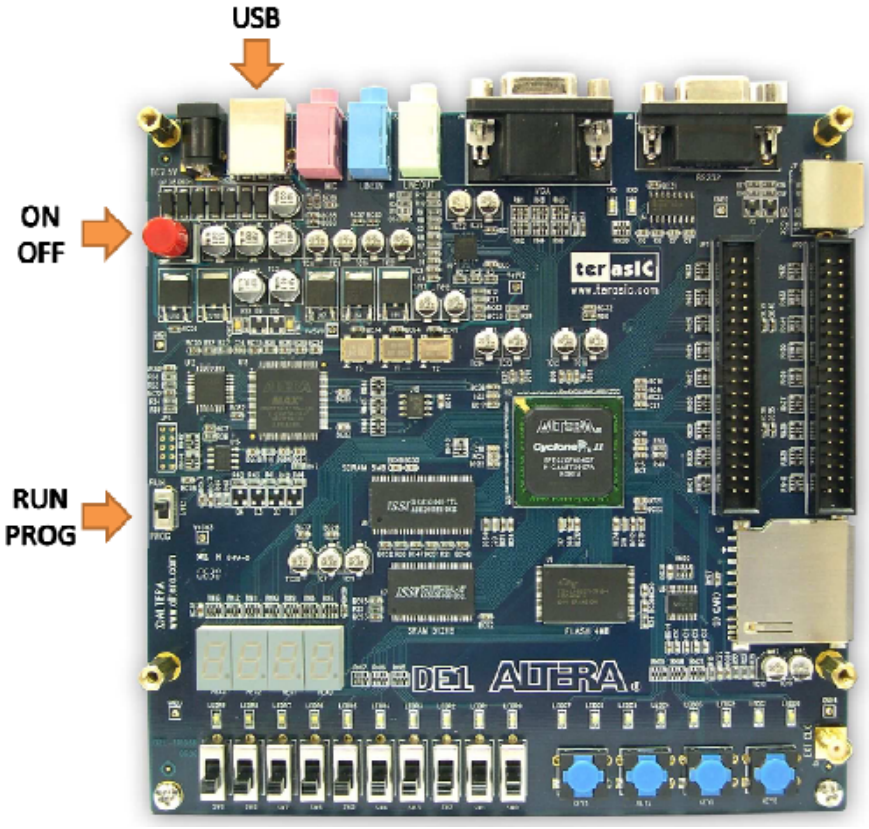

Figura 13. Tarjeta de lógica programable del laboratorio.

A continuación hay que arrancar la opción *Programmer* del *QuartusII*. En el Menú principal seleccionar `Tools→Programmer`. Aparece entonces la ventana siguiente, Figura 14, que contiene la selección del fichero y el puerto por donde se programa.

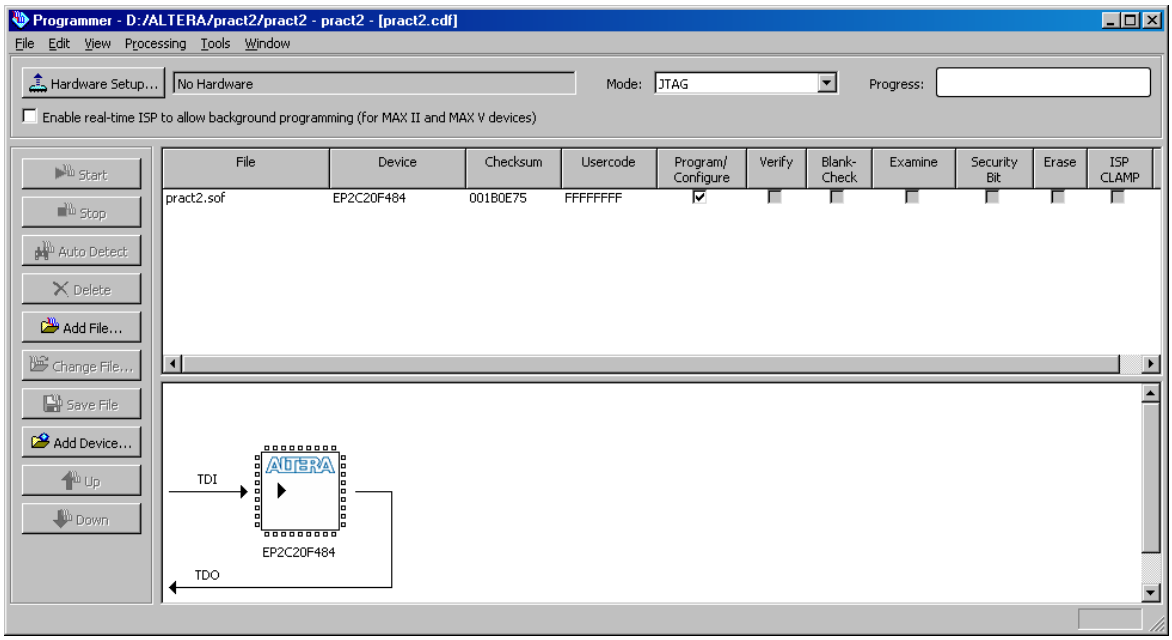

Figura 14. Ventana de programación del dispositivo.

Antes de continuar hay que verificar que el programa ha reconocido el hardware de programación. Para ello hay que mirar si en la parte superior izquierda de la ventana del programador pone *USB-Blaster [USB-0]* o si pone otra cosa distinta, como *No Hardware* (tal y como se muestra en la figura anterior). En este último caso hay que configurar el hardware correcto, y para ello hay que pulsar en el botón `Hardware Setup...`, con lo que aparecerá la ventana de la Figura 15. A continuación hay que hacer doble clic en *USB-Blaster* de la lista *Availabe Hardware items* y comprobar que en el campo `Currently selected hardware` aparece USB-Blaster [USB-0]. Una vez hecho esto hay que pulsar el botón `Close`.

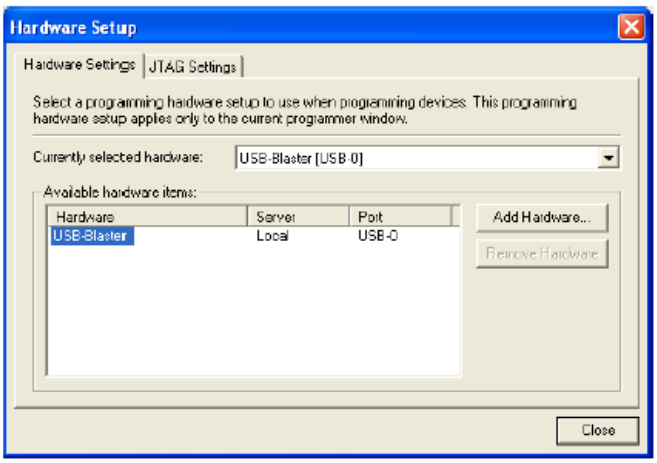

Figura 15. Ventana de configuración del dispositivo.

Por último, hay que seleccionar la casilla de la columna `Program/Configure`, tal y como se muestra en la Figura 14), y pulsar el botón `Start` para que se descargue el circuito diseñado en la FPGA. Es buen momento también para verificar que el dispositivo seleccionado es el correcto, debajo de la opción

Device. Si todo ha ido bien, el circuito está descargado y puede comenzar a evaluarlo. Si ha habido algún error, lo muestra en la misma ventana de mensajes del *QuartusII*.

Una vez configurado el circuito en la FPGA hay que probar TODAS las combinaciones posibles de las entradas para verificar si su funcionamiento es correcto. Si se detecta algún error hay que corregir el esquema capturado en la práctica anterior, guardar los cambios realizados, compilar, simular y volver a volcar a la placa, comprobando el correcto funcionamiento en cada paso del proceso.

Una vez que el funcionamiento del circuito es correcto en todos los pasos, muestre el resultado al profesor.